基于变迁图编辑距离的流程相似性算法 *

段 瑞，方 欢，方贤文，詹 悦

(安徽理工大学 数学与大数据学院, 安徽 淮南 232001)

摘　要：为了提高从企业模型库中查询检索模型的效率，提出一种基于变迁图编辑距离的流程相似性算法。首先，给出变迁图的概念及其生成方法；其次，提出边的长度概念，删除和插入边的代价由该边的长度决定，基于此定义图编辑操作及其代价，并用节点匹配算法计算最小图编辑距离；然后，给出两个过程模型的相似性概念和计算方法；最后，通过实验验证了算法的正确性且满足七条相似性性质，并验证了变迁图编辑距离满足四条距离性质。

关键词：Petri 网；相似性度量；变迁图；图编辑距离

中图分类号：TP301　　doi: 10.19734/j.issn.1001-3695.2018.10.0729

Process similarity algorithm based on editing distance of transition graph

Duan Rui, Fang Huan, Fang Xianwen, Zhan Yue

(*School of Mathematics & Big Data*, *Anhui University of Science & Technology*, *Huainan Anhui* 232001, *China*)

Abstract: In order to improve the efficiency of querying and retrieving models from the enterprise model library, a process similarity algorithm based on the edit distance of transition graph is proposed. Firstly, the concept of the transition graph and its generation method are given. Secondly, the concept of the length of the edge is proposed. The cost of deleting and inserting the edge is determined by the length of the edge. Based on this, the graph editing operation and its cost are defined, and the node matching algorithm is used to calculate the minimum graph editing distance. Then, the similarity concept and calculation method of the two process models are given. Finally, the correctness of the algorithm is verified and the seven similarity properties are satisfied, and the editing distance of the transition graph is verified to satisfy the four distance properties.

Key words: Petri nets; similarity measure; transition graph; graph editing distance

0 引言

业务流程作为企业的三要素之一，流程模型的相似性度量一直是工作流研究中的一个重要方向[1]。对于数以万计的模型库，高效且精准的模型检索方法成为业务过程管理的关键，这就要求高效的流程相似性算法[2]。

目前已有的流程相似性算法主要基于以下三个方面展开[3]：a)最直观的相似性度量方法，关注任务标签、事件或其他建模元素；b)从模型的拓扑结构出发，关注模型的元素集合及元素之间的行为关系；c)基于行为语义的相似性算法。为了验证算法的性能，流程相似性领域学者和专家提出了七条基本的相似性性质[4~6]：顺序结构漂移不变性、并发结构漂移不变性、循环结构漂移不变性、互斥结构漂移不变性、跨度负相关性、非替代无关递减性和循环序列长度负相关性。

Zha 等人[7]提出了一种基于变迁紧邻关系的相似性算法，称为 *TAR* 算法，该算法通过考察流程变迁的两两紧邻关系来表征流程模型的相似性。殷明等人[5]提出了一种 *TAR* 算法的改进算法 *TAR++* 算法，创造性的为 *TAR* 增加了重要性系数，有效地满足了一些 *TAR* 算法不能满足的相似性性质。但 *TAR++* 算法用深度优先搜索方法为紧邻变迁关系分配重要性，算法的最坏时间复杂度为 $O(V+E+N!)$，是一个阶乘级的复杂度，且 *TAR++* 相似性有一个不高的上界。

Weidlich 等人[8]提出了基于行为轮廓(*BP*)的相似度度量方法，该方法定义了弱序，并基于弱序关系提出一系列关系，统称为行为轮廓，该方法用行为轮廓扩展紧邻变迁关系，有效地满足了一些 *TAR* 算法不能满足的相似性性质，但仍不能满足所有性质。通过构造任务最短跟随距离矩阵度量流程相似性的 *SSDT* 算法[4]能够满足所有相似性性质，但是 *SSDT* 算法每次计算流程相似性时，需要根据矩阵的秩进行对应的扩展确保两个矩阵的秩相等，使得算法性能不够优良。

图匹配和图编辑距离一直是模型相似性度量的重要手段，Dijkman 等人[9]描述了四种图匹配算法，分别是贪心算法、带剪枝的穷举算法、流程启发式算法及 A^* 算法，并通过实验评估表明：贪心算法所耗时间远低于其他算法，A^* 算法的精确度最高，带剪枝的穷举算法所耗时间远高于其他算法。图编辑距离是指一个图转换成另一个图需要的操作代价[10]，计算流程相似性需要的是最小图编辑距离，图匹配算法即通过在模型比较时建立节点之间一一对应关系，从而找到最小图编辑距离。

针对相似性度量算法仍存在的一些问题，如不能满足相似性性质、时间复杂度较高、空间爆炸、灵活性较低、计算值与预期值不符等，本文提出一种新的基于变迁图编辑距离的流程相似性算法 *TGED*，主要贡献为：a)通过库所映射，把 Petri 网模型转换成变迁图，即把 Petri 网模型中的库所映射

收稿日期：2018-10-01；**修回日期：**2018-11-21　**基金项目：**国家自然科学基金资助项目（61472003，61402011，61572035）；安徽省自然科学基金资助项目（1608085QF149）；安徽省高校优秀青年人才基金资助项目（gxyqZD2018038）；安徽省博士后基金资助项目（2018B288）

作者简介：段瑞（1993-），男，安徽涡阳人，硕士，主要研究方向为 Petri 网理论与应用（85768312@qq.com）；方欢（1982-），女，安徽合肥人，副教授，博士，主要研究方向为业务流程管理系统；方贤文（1975-），男，河南信阳人，教授，博士，主要研究方向为 Petri 网与可信软件；詹悦（1994-），女，安徽太湖人，硕士，主要研究方向为 Petri 网理论与应用.

成变迁图中的边；b)首次提出边的长度概念，编辑不同长度的边所需代价不同；c)取一个变迁图到另一个变迁图所需操作的最小代价为编辑距离，并基于此计算相似性；d) *TGED* 算法有一个较大的上界，且时间复杂度较低。

1 预备知识

本文以 Petri 网作为形式化的建模和分析工具，在介绍 *TGED* 算法之前，先介绍关于 Petri 网的预备知识。

定义 1　标签 Petri 网。满足下列条件的一个五元组 $LN=(P,T,F,\Sigma,\ell)$ 称做标签 Petri 网，其中：P 是库所；T 是变迁；F 是流关系；Σ 是表示变迁的标签集合。

a)　$P\cup T\neq\varphi$；

b)　$P\cap T=\varphi$；

c)　$F\subseteq(P\times T)\cup(T\times P)$；

其中：$\ell:T\rightarrow\Sigma\cup\{\varepsilon\}$ 是标签函数。

记 $X=P\cup T$，对于 $x\in X$，x 的前集记为 $\bullet x=\{y\in X\mid(y,x)\in F\}$，$x$ 的后集记为 $x\bullet=\{y\in X\mid(x,y)\in F\}$，$\bullet(x\bullet)=\{z\in X\mid y\in X\wedge(x,y)\in F\wedge(z,y)\in F\}$。$x\in X$ 称为 Petri 网的一个节点，$f\in F$ 称为 Petri 网的一个边，与 x 相邻的边称为 x 的边，包括入边和出边。为标签 Petri 网添加一个初始标志 M_0 得 $LN=(P,T,F,M_0,\Sigma,\ell)$ 称为标签 Petri 网系统，$M_0:P\rightarrow N^+$，N^+ 为非负整数集。

定义 2　工作流网。满足下列条件的三元组 $WFN=(N,i,o)$ 称做工作流网，其中：$N=(P,T,F,M_0,\Sigma,\ell)$ 是标签 Petri 网系统；$i\in P$ 是开始库所；$o\in P$ 是结束库所。

a) i,o 是唯一的；

b) $\bullet i=\varphi$，$o\bullet=\varphi$，$i\neq o$；

c) $\forall x\in P\cup T$，存在一条从 i 到 o 的路径包含 x。

本文所讨论的模型都是基于 Petri 网的安全工作流网，称所有库所都是有界且界为 1 的 Petri 网为安全的，称库所最多含有一个标志为有界且界为 1 的，其中初始标志为整个系统模型的触发条件，初始标志状态下，开始库所中含有一个标志，其他库所不含标志。为了方便计算相似性，提出一种不含库所只由变迁和边组成的图，称为变迁图。

定义 3　图编辑距离。给定两个图 $G_1=(V_1,E_1)$ 和 $G_2=(V_2,E_2)$，由 G_1 转换成 G_2 所需操作的最小代价称为图 G_1 和 G_2 的编辑距离，记为 $edis(G_1,G_2)$。

为了确定图编辑距离，需要定义合理的图编辑操作及其代价，本文总结已有的图编辑操作，并给出一个新的概念：边长度，删除和插入边由边的长度决定。

定义 4　图编辑操作。给定图 $G=(V,E)$，定义编辑图 G 的基本操作为：节点的删除、插入、替换及边的删除、插入。$(u\rightarrow\varepsilon)$ 表示删除节点 u，$(\varepsilon\rightarrow v)$ 表示插入节点 v，$(u\rightarrow v)$ 表示用节点 v 替换节点 u，同理表示边的删除和插入。

如图 1 所示，从图 1(a)到(b)为删除边 (a,f) 和 (f,c)，从(b)到(c)为用节点 g 替换节点 f，即 $(g\rightarrow f)$，从(c)到(d)为插入边 (g,b)、(g,d) 和 (g,e)，经过一系列基本的图编辑操作，图 1(a)转换成(d)。

2基于变迁图的相似性计算

为了计算两个流程模型的相似性，首先需要依据它们的 Petri 网模型生成变迁图，其次为变迁图定义合理的基本图编辑操作及其代价，最后求得最小代价即为图编辑距离。

2.1变迁图的生成

定义 5　变迁图。给定工作流网 $WFN=(N,i,o)$，其中 $N=(P,T,F,M_0,\Sigma,\ell)$ 是标签 Petri 网系统；$G=(V,E)$ 是 *WFN* 的变迁图，V 是节点集合且 $\bigcup V=T$，$E\subseteq V\times V$ 是边集合。E 的产生方式为：$E=\{(\bullet p,p\bullet)\mid\forall p\in P\}$，记边 $(\bullet p,p\bullet)$ 的前端节点为 $\bullet p$，后端节点为 $p\bullet$。特殊地，若 $\exists p\in P[\mid\bullet p\mid>1]$，$p$ 映射成的边的前端节点由前面边的后端节点确定，p 或作为互斥结构的收尾，对应 $|\bullet p|$ 条边，或引发一个循环结构；若 $\exists p\in P[\mid p\bullet\mid>1]$，$p$ 引发一些互斥分支，若 $p_1,p_2,\ldots,p_j\in p\bullet\wedge p_1\bullet\bullet\neq p_2\bullet\bullet\neq\ldots\neq p_j\bullet\bullet$，则 p 对应边的后端节点将引出 j 条边，且每条边由一个该节点中的变迁控制，该变迁称为控制变迁。

为了处理工作流网首尾是特殊结构的情况，在生成变迁图之前，人工为工作流网增加开始库所 p_s 和变迁 t_s、结束库所 p_e 和变迁 t_e，并增加四条边分别为 (p_s,t_s)、(t_s,i)、(o,t_e)、(t_e,p_e)。

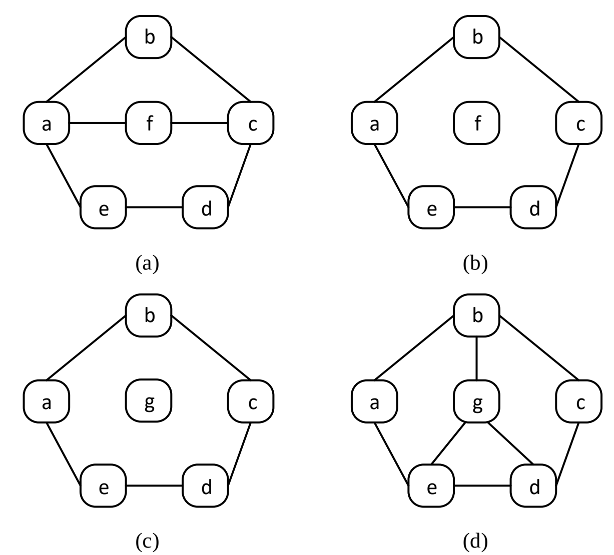


图 1　图编辑操作示例(a)~(d)

Fig. 1　Graph editing operation example (a)~(d)

如图 2 所示，左边是 4 种典型的 Petri 网模型结构：顺序、互斥、循环和并发，右边是对应的变迁图。图 2(a)左边是顺序结构，人工为其增加开始库所 p_s 和变迁 t_s、结束库所 p_e 和变迁 t_e，并增加 4 条边分别为：(p_s,t_s)、(t_s,i)、(o,t_e)、(t_e,p_e)，新得到的模型除库所 p_s 和 p_e 外有 4 个库所，依据定义 5 对应 4 条边，得到如图 2(a)的右边。图 2(b)左边是内嵌有顺序结构的互斥结构，人工改造后：对于库所 s_1，$s_1\bullet=\{t_2,t_3\}$，即 $s_1\in P[\mid s_1\bullet\mid>1]$，且 $t_2\bullet\bullet=\{t_4\}$、$t_3\bullet\bullet=\{t_e\}$，因此有 $t_2\bullet\bullet\neq t_3\bullet\bullet$，库所 s_1 对应边 $(\{t_1\},\{t_2,t_3\})$ 的后端节点 $\{t_2,t_3\}$ 将引出两条边，分别由 t_2、t_3 控制，如图 2(b)右边。图 2(c)左边是循环结构，对于库所 s_1，$\bullet s_1=\{t_1,t_4\}$，则 s_1 对应的边的前端节点由库所 i 对应的边的后端节点决定，即 $\{t_1\}$，此时 s_1 引发一个循环结构。图 2(d)左边是并发结构，在本文给出的变迁图定义中，顺序结构和并发结构是最易处理的结构，图 2(d)右边是其对应的变迁图。

2.2变迁图的行为语义

变迁图作为 Petri 网模型的无库所简化，每个节点可能含有多个变迁，因为一个库所能够引发多个互斥分支。同理，变迁图中一个节点可能具有多条边：a)作为嵌有其他结构的互斥结构的收尾；b)引出的互斥分支具有循环结构；c)并发结构。本文称一条从人工添加变迁 t_s 到 t_e 的完整执行序列为一条语句，一个变迁图的所有语句组成变迁图的语言。对变迁图的行为保持解释如下：

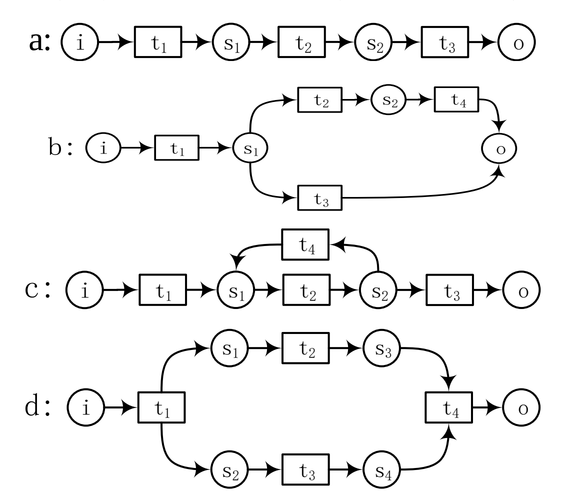
a)变迁图的所有发生序列即为变迁图的行为语义。

变迁图中的边由变迁控制，边上的变迁决定了该边只能

由控制变迁引发，如图 2(b)中的变迁图的发生序列为：$<\{t_s\},\{t_1\},\{t_2,t_3\},\{t_4\},\{t_e\}>$ 和 $<\{t_s\},\{t_1\},\{t_2,t_3\},\{t_e\}>$，若边上为空，则默认控制变迁为该边的前端节点元素；

变迁图中只含有一个变迁的节点引出的多条边并发，可以由上一条解释，即该节点引出的边默认由该节点元素控制，由于节点只含有一个变迁，当该变迁被引发之后，即可以引发所有从其引出的边，如图 2(d)中变迁图的发生序列为：$<\{t_s\},\{t_1\},\{t_2\},\{t_3\},\{t_4\},\{t_e\}>$ 和 $<\{t_s\},\{t_1\},\{t_3\},\{t_2\},\{t_4\},\{t_e\}>$；

b)变迁图中节点触发条件：以该节点为后端节点的边被引发。



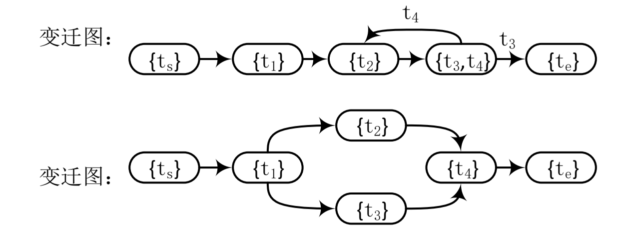


图 2　四种典型的 Petri 网模型结构(a)~(d)及对应的变迁图

Fig. 2　There are 4 typical Petri net model structures on the left (a)~(d),corresponding transition graph on the right

2.3图编辑操作及其代价

前面提到，本文给出的图编辑基本操作有：节点的删除、插入、替换及边的删除、插入，本节主要为这些基本操作赋予合理的代价。本文认为节点的删除、插入为单位基本操作，包括控制变迁的插入、删除和替换，赋予代价 1。下面详细介绍节点的替换和边的删除、插入。

2.3.1节点的替换

节点的替换涉及到节点中变迁标签的相似性，标签的单位字删除、插入和替换代价为 1，通过一系列字符串操作把一个变迁标签转换成另一个变迁标签，所需操作的最小代价为两个变迁标签的编辑距离，与它们中模值最大的比值为节点的替换操作代价。若两个变迁标签完全不同，则替换操作代价为 1。

如图 3(a)中变迁标签"*Looking for food*"到图 3(b)中变迁标签"*Find food*"的转换所需最小操作代价为 10，即替换 4 个单位字，删除 6 个单位字，整个节点的替换代价为 $\frac{10}{14}=0.714$。

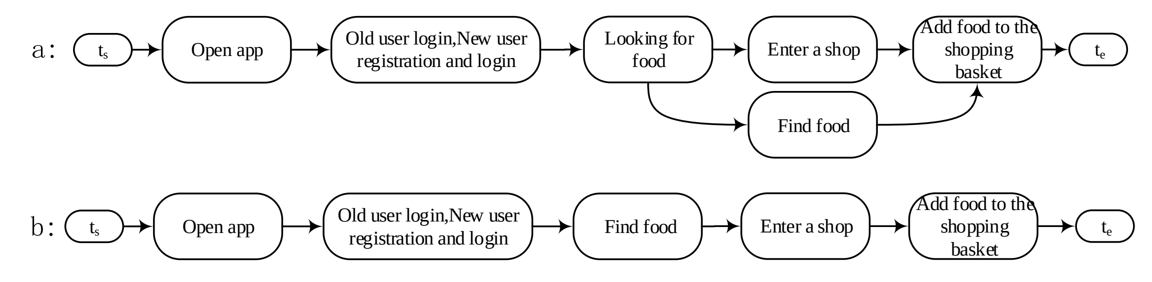


图 3　某美食 app 部分系统模型对应的变迁图(a)~(b)

Fig. 3　Transition graph (a)~(b) corresponding to the system model of gourmet app

2.3.2边的删除与插入

本文首先提出边的长度概念用以描述删除与插入该边的代价，库所 *i* 和 *o* 对应的边长度为 1，在人工变迁 t_s 与 t_e 之间选取一条最长简单路径为主干，主干上的边长度均设为 1，主干的分支跨主干的长度为该分支的近似长度，节点长度为 0。

如图 3 所示，图 3(a)转换到(b)除用"*Find food*"替换"*Looking for food*"之外，还需删除节点"*Find food*"及两条边({"*Looking for food*"},{"*Find food*"})、({"*Find food*"},{"*Add food to the shipping basket*"})。当节点"*Find food*"和节点"*Enter a shop*"无限接近时，取这两个边的长度近似值 1，则删除代价为 1。

2.4相似性计算

由定义 3 可知，一个图转换成另一个图所需操作的最小代价为它们的编辑距离，基于变迁图编辑距离的相似性度量方法的核心是求变迁图的最小编辑距离。

定义 6　*TGED* 相似性。给定两个工作流网，它们对应的变迁图分别为 $G_1=(V_1,E_1)$ 和 $G_2=(V_2,E_2)$，$edis(G_1,G_2)$ 为图 G_1 和 G_2 的编辑距离，$|G|=|G.V|+|G.E|+|G.CT|$ 表示图 G 的模值，其中 $|G.V|$ 表示图 G 中的节点数，$|G.E|$ 表示图 G 中所有边的近似长度之和，$|G.CT|$ 表示图 G 中控制变迁数，则相似性为 $sim(G_1,G_2)=1-\frac{edis(G_1,G_2)}{\max(|G_1|,|G_2|)-2}$。

对于定义 6 的相似性计算公式中分母减去 2 是因为要消除人工添加的变迁的影响，如图 3 所示，$edis(a,b)=3.714$，则 $sim(a,b)=1-\frac{edis(a,b)}{\max(|a|,|b|)}=1-\frac{3.714}{16-2}=0.735$。

3TGED 算法

3.1算法设计

本节给出基于变迁图编辑距离的流程相似性算法 *TGED*。由前面可知，算法的核心是生成变迁图及计算变迁

图编辑距离，依据变迁图编辑距离可以轻松的计算出相似性。

分析 *TGED* 算法，算法第 1 行人工改造工作流网，第 2~16 行为改造后的工作流网生成变迁图，其中 2~4 行创建并初始化变迁图，5~15 依据定义 5 计算工作流网的变迁图，16 行计算另一个工作流网的变迁图，第 17 行计算两个变迁图的最小编辑距离，第 18~19 行依据定义 6 计算相似性。

在 *TGED* 算法第 17 行计算两个变迁图转换所需的最小操作代价不是一个简单的过程，文献[6]提出了基于 A^* 搜索算法的节点匹配，在算法迭代过程中，不断利用当前生成的部分进行节点匹配，选择代价最小的(相似性最大)进行扩展，最终得到最优解，该算法定义了剩余节点的编辑距离下界估值，用以“剪枝”。文献[9]给出了 4 种匹配算法用以解决“诱导最大相似性的映射”，即最小图编辑距离，依据 4 种算法的平均精确度和运行时间，本文采用贪心算法和 A^* 算法。

$transition-graph-edit-distance-similarity-algorithm(W_1,W_2)$

1. $artificial\ \ transformation\ workflow\ nets\ W_1,W_2$
2. $create\ graph\ G_1=(V_1,E_1)$
3. $V_1=\varphi$
4. $E_1=\varphi$
5. $foreach\quad in\ a\ certain\ order\ s\in P_1$
6. $\quad if\quad |\bullet s|=1$
7. $\qquad V_1=V_1\cup\bullet s\cup s\bullet$
8. $\qquad E_1=E_1\cup(\bullet s,s\bullet)$
9. $\quad if\quad |\bullet s|>1$
10. $\qquad the\ place\ in\ front\ of\ s\ determine\ s$
11. $\qquad V_1=V_1\cup s\bullet$
12. $\qquad foreach\quad s'\subseteq\bullet s\wedge\exists node\ s''\in V_1[(s'',s')\in E_1]$
13. $\qquad\quad E_1=E_1\cup(s',s\bullet)$
14. $\quad if\quad |s\bullet|>1\wedge\exists s_1,s_2,..,s_j\in s\bullet\wedge s_1\bullet\bullet\neq s_2\bullet\bullet\neq...\neq s_j\bullet\bullet$
15. $\qquad incorporate\ the\ j\ edges\ of\ node\ s\bullet\ into\ E_1$
16. $same\ as\ above,calculate\ transition\ graph\ G_2\ of\ W_2$
17. $calculate\ minimum\ operating\ cost,recorded\ as\ edis(G_1,G_2)$
18. $calculate\ the\ modulus\ of\ G_1\ and\ G_2,recorded\ separately\ |G_1|,|G_2|$
19. $similarity\ between\ W_1\ and\ W_2,sim(W_1,W_2)=1-\dfrac{edis(G_1,G_2)}{\max(|G_1|,|G_2|)}$

3.2算法时间复杂度分析

改造工作流网的方法为：人工添加两个变迁、两个库所及四条边，时间复杂度为常数量级；创建变迁图及初始化的时间复杂度同样为常数量级；*TGED* 算法第 5~16 行生成变迁图的时间复杂度分别为 $O(|V_1|+|E_1|)$ 和 $O(|V_2|+|E_2|)$，用摊还分析即可得到该时间复杂度；用贪心算法计算最小图编辑距离的时间复杂度为 $O(\min(|V_1|,|V_2|)\times|V_1|\times|V_2|)$，$A^*$ 算法计算最小图编辑距离的最坏情况下时间复杂度为 $O(\min(|V_1|!,|V_2|!))$，最佳情况下时间复杂度为 $O(|V_1|\times|V_2|)$。实际运行时，由于使用了估值函数，A^* 算法的运行时间远低于最坏时间复杂度，接近最佳时间复杂度。

4实验与评估

本文的实验模型主要来自 SAP 模型库，从 SAP 模型库中随机获取 230 个模型作为实验对象。为了验证 *TGED* 算法相似性性质的满足情况，另人工编纂 70 个模型，一共 300 个流程模型作为实验数据。

4.1实验设计

本文的实验机器环境为：Intel(R) Core(TM) i5-7300HQ CPU @ 2.50 GHz，安装内存(RAM)为 8.00 GB，64 位操作系统。以开放的业务过程模型管理框架 BeehiveZ 系统[11,12]为工具，BeehiveZ 具有多种流程管理功能，本文主要用到查询功能，即检索功能。实验数据共分为两类：第一类是从 SAP 模型库中随机获取的 230 个模型，用以验证 *TGED* 算法的正确性和可行性；第二类是人工编纂的 70 个模型，用于验证 *TGED* 算法相似性性质的满足情况。

首先通过实验验证 *TGED* 算法的正确性，即 *TGED* 算法是否能在输入两个工作流网的情况下，输出一个 $[0,1]$ 之间的数；其次需要验证图编辑距离性质，以三角不等式性质为主；最后验证 *TGED* 算法相似性性质满足情况。

4.2距离性质验证

对本文提出的图编辑距离进行距离性质验证，*TGED* 相似性度量满足 4 个距离性质，分别为对称性、自反性、非负性、三角不等式性。

对称性：两个工作流网之间的相似性唯一，即 $\forall W_1,W_2[edis(W_1,W_2)=edis(W_2,W_1)]\Rightarrow sim(W_1,W_2)=sim(W_2,W_1)$，图编辑操作都是对称的，所以两个模型之间的最小编辑操作代价是对称的，它们的 *TGED* 相似性满足对称性。

自反性：工作流网和自身的相似性为 1，即 $\forall W[edis(W,W)=0]\Rightarrow sim(W,W)=1$，两个完全相同的图不需要任何操作便能互相转换。

非负性：两个图的 *TGED* 相似性计算为：$sim(G_1,G_2)=1-\dfrac{edis(G_1,G_2)}{\max(|G_1|,|G_2|)-2}$，其中 $edis(G_1,G_2)\le\max(|G_1|,|G_2|)-2$ 恒成立，即 $\forall W_1,W_2[sim(W_1,W_2)\ge 0]$。

三角不等式：$\forall W_1,W_2,W_3[edis(W_1,W_2)\le edis(W_1,W_3)+edis(W_3,W_2)]$，即给定任意 3 个模型及它们之间的 3 个距离，任意两个距离之和大于等于第三个距离，这是 *TGED* 相似性最重要的一条距离性质，也是需要实验验证的。

首先给出三角不等式满足率的定义，假设实验数据集中共有 n 个模型，从这 n 个模型中任取 3 个，共有 C_n^3 种组合，若共有 n' 种模型组合满足三角不等式，则三角不等式满足率为 n'/C_n^3。其次为了更好的对不同算法的三角不等式满足率进行评估，本文采取 3 次实验，得到 3 组三角不等式满足率，每次从模型数据集中随机抽取一定数量的模型作为实验数据集，3 次实验抽取的模型数量分别为 94、117 和 123，3 次实验数据集分别记作数据集 1、2 和 3。最后为了更好地体现 *TGED* 算法的优势，除引言部分提到的相似性算法外，本节额外加入两个算法作为实验对象，分别为：因果足迹算法 *CF* [10]和完整触发序列算法 *CFS* [13]。

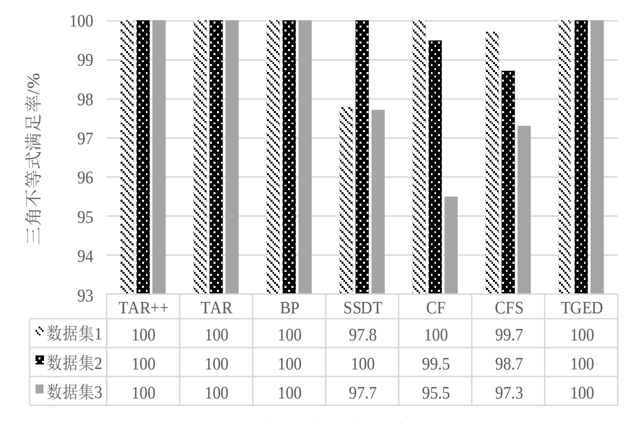
前面提到，插入和删除边的代价由该边的长度决定，为了精确验证三角不等式满足率，在计算图编辑距离时，插入和删除边的代价便不能取近似值。本文用 "$n+$" 表示比长度 "n" 大一点点的长度，即插入和删除他们的代价。

对 $TAR++$、*TAR*、*BP*、*SSDT*、*CF*、*CFS* 算法和本文提出的 *TGED* 算法进行三角不等式满足率实验，表现如图 4 所示。在三角不等式满足率方面，*SSDT*、*CF* 和 *CFS* 算法表现得不如其他算法，*TAR* 算法在数据集 1 上未达到 100%满足率，表现最好的是 $TAR++$、*BP* 和 *TGED* 算法。

更全面的算法性能比较还包括算法运行时间的对比，如图 5 所示，*TGED* 算法分为用贪心算法进行节点匹配的 *TGED(G)* 算法和用 A^* 算法进行节点匹配的 *TGED(A)* 算法。在计算相似性花费时间方面，*CFS* 算法明显高于其他算法，*TGED(G)* 算法所需时间比其他算法都要少，但在准确率方面低于 *TGED(A)* 算法。结合三角不等式满足率得出结论：*TGED* 算法在整体性能表现上略优于其他算法。

在三角不等式满足率上贪心算法略差于穷举算法和 A^* 算法，但在运行时间上有一定的优势。因此，在对精确度要求很高的情况下，选择 A^* 算法进行节点匹配，在对精确度要

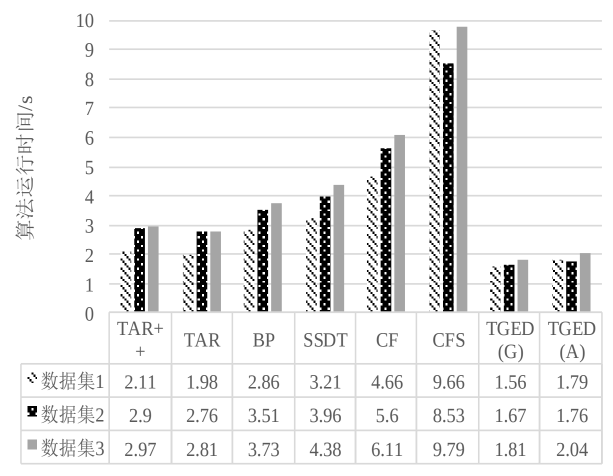
求不高的情况下，则选择贪心算法以节省时间。



	TAR++	TAR	BP	SSDT	CF	CFS	TGED
数据集1	100	100	100	97.8	100	99.7	100
数据集2	100	100	100	100	99.5	98.7	100
数据集3	100	100	100	97.7	95.5	97.3	100

图 4　三角不等式满足率对比

Fig. 4　Triangle inequality satisfaction rate



	TAR++	TAR	BP	SSDT	CF	CFS	TGED (G)	TGED (A)
数据集1	2.11	1.98	2.86	3.21	4.66	9.66	1.56	1.79
数据集2	2.9	2.76	3.51	3.96	5.6	8.53	1.67	1.76
数据集3	2.97	2.81	3.73	4.38	6.11	9.79	1.81	2.04

图 5　算法运行时间对比

Fig. 5　Algorithm running time comparison

4.3相似性性质验证

目前已提出的相似性性质主要有顺序结构漂移不变性、并发结构漂移不变性、互斥结构漂移不变性、循环结构漂移不变性、跨度负相关性、非替代无关递减性及循环序列长度负相关性。实验数据模型集为人工编纂的 70 个流程模型。

性质 1　顺序结构漂移不变性。无论在顺序结构中哪部分插入新变迁得到新的顺序结构模型，新模型与原模型的相似性均相等。

人工制造一个含有 10 个变迁的顺序结构模型作为原模型 N，如图 6 所示，限于篇幅中间略去部分变迁，在原模型的变迁之间插入新变迁，得到 11 个新模型，图略。把所有模型转换成变迁图，则每个新模型变迁图到原模型变迁图的编辑距离均为 2，即所有新模型与原模型的相似性均为 0.913。

性质 2　并发结构漂移不变性。对流程模型中的顺序结构进行改造，改造成的并发结构分支无论添加在顺序结构的哪部分上，得到的新模型与原模型的相似性均相等。

性质 3　互斥结构漂移不变性。对流程模型中的顺序结构进行改造，改造成的互斥结构分支无论添加在顺序结构的哪部分上，得到的新模型与原模型的相似性均相等

性质 4　循环结构漂移不变性。对流程模型中的顺序结构进行改造，改造成的循环结构分支无论添加在顺序结构的哪部分上，得到的新模型与原模型的相似性均相等

其实，前 4 条性质可以总结为 1 条性质，即结构漂移不变性，但是有的算法只能满足前 4 条性质中的一部分，为了加以区分，分开叙述前 4 条性质。图 7 中模型 N_1、N_2、N_3 分别为按性质 2、3、4 改造原模型得到的新模型，抽取其中一个在图 7 中表示，其余不作赘述。对于性质 2，无论并发结构分支添加在哪部分上，对应的变迁图编辑距离均为 3；对于性质 3，无论互斥结构分支添加在哪部分上，对应的变迁图编辑距离均为 0.5；对于性质 4，无论循环结构分支添加在哪部分上，对应的变迁图编辑距离均为 3.5。其中，代价为 0.5 的操作均是出自节点替换，即在含一个变迁的节点中插入另一个变迁所需代价。

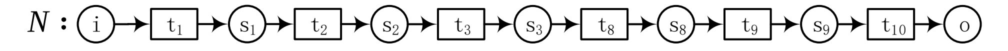


图 6　人工编纂的原模型,共含有 10 个变迁其中略去一部分

Fig. 6　Manually complied original model,there are 10 transitions and skip part

图 7　按性质 2、3、4 改造原模型得到的新模型

Fig. 7　New models obtained by modifying original model according to properties 2,3and4

图 8　相比于 N_2 跨度更大的互斥结构

Fig. 8　Mutual exclusion structure larger than span of　N_2

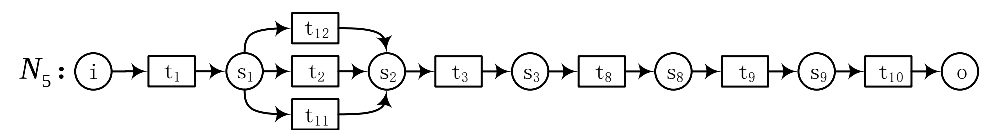


图 9　非替代无关递减性

Fig. 9　Non-replacement-independent decline

性质 5 跨度负相关性。对模型中顺序结构添加互斥结构分支，该分支在原模型结构上跨度越大，得到的新模型与原模型的相似性越小。

如图 8 所示，对 N 添加跨度为 2 的互斥结构得到模型 N_4，$sim(N_4, N) = 0.820$，由前面可知 $sim(N_2, N) = 0.976$，有 $sim(N_4, N) < sim(N_2, N)$，增加互斥结构的跨度，得到另外一系列模型，验证了跨度负相关性的正确性。

性质 6 非替代无关递减性。对模型中顺序结构添加互斥结构分支，添加的分支越多，得到的新模型与原模型的相似性越小。

性质 7 循环序列长度负相关性。对流程模型中的顺序结构进行添加循环结构改造，循环结构的跨度越大，得到的新模型与原模型的相似性越小。

如图 9 所示，模型 N_5 为在 N 上同一部分添加两个互斥分支，在 N_2 上互斥结构部分再添加一个互斥分支，有 $edis(N_5, N) > edis(N_2, N)$，即 $sim(N_5, N) < sim(N_2, N)$。性质 7 与性质 5 类似，实验验证 *TGED* 相似性满足所有 7 种相似性性质。

表 1 为各算法对 7 种相似性性质的满足情况，√表示满足，×表示不满足。由表 1 可知，*TAR++*、*SSDT*、*CFS* 算法和 *TGED* 算法(本文算法)满足所有相似性性质，*SSDT* 算法每次计算流程相似性的时候，需要根据矩阵的秩进行对应的扩展确保两个矩阵的秩相等；*CFS* 算法会对并发任务导致的所有可能执行序列进行逐一枚举。因此，和 *CFS* 算法运行时间过长。

TAR++ 算法运行时间稍长于 *TGED* 算法，但 *TAR++* 算法用深度优先搜索方法为紧邻变迁关系分配重要性，算法的最坏时间复杂度为 $O(V + E + N!)$，最坏情况往往会出现。结合各算法的三角不等式满足率和运行时间进一步得出结论：*TGED* 算法略优于其他算法。

表 1　各算法对相似性性质的满足情况

Table 1　Satisfaction of similarity properties of each algorithm

	性质 1	性质 2	性质 3	性质 4	性质 5	性质 6	性质 7
TAR++ [5]	√	√	√	√	√	√	√
TAR [7]	×	√	×	√	×	√	×
BP [8]	√	√	√	√	×	√	√
SSDT [4]	√	√	√	√	√	√	√
CF [10]	√	√	×	×	×	√	√
CFS [13]	√	√	√	√	√	√	√
TGED(本文方法)	√	√	√	√	√	√	√

5 结束语

为了提高从企业模型库中查询和检索模型的效率，解决已有流程相似性算法存在的一些问题，本文提出基于变迁图编辑距离的流程相似性算法。该算法通过把模型转换成简单的变迁图，计算最小变迁图编辑距离，得出相似性。另外，本文首次提出了边的长度概念，删除和插入边的代价由边的长度决定，基于此定义了图编辑操作代价，并用贪心算法和 A^* 算法计算最小图编辑距离。通过实验证明了本文算法整体性能上略优于其他算法。

未来，希望通过深入研究各种相似性度量方法，扩展相似性性质，优化 *TGED* 算法或提出新的更好的算法，使相似性更加贴近领域专家的评估结果。

参考文献：

[1]Lu Yahui, Yu Haofei, Ming Zhong, *et al.* A similarity measurement based on structure of business process [C]// Proc of IEEE International Conference on Computer Supported Cooperative Work in Design. 2016: 498-503.

[2]Montani S, Leonardi G, Quaglini S, *et al.* A knowledge-intensive approach to process similarity calculation [J]. Expert Systems with Applications, 2015, 42 (9): 4207-4215.

[3]Wang Jianmin, Jin Tao, Wong R K, *et al.* Querying business process model repositories [J]. World Wide Web-internet & Web Information Systems, 2014, 17 (3): 427-454.

[4]汪抒浩，闻立杰，魏代森，等. 基于任务最短跟随距离矩阵的流程模型行为相似性算法 [J]. 计算机集成制造系统，2013, 19 (8): 1822-1831. (Wang Shuhao, Wen Lijie, Wei Daiseng, *et al.* SSDT matrix-based behavior similarity algorithm for process model [J]. Computer integrated manufacturing system, 2013, 19 (8): 1822-1831.)

[5]殷明，闻立杰，王建民，等. 基于变迁紧邻关系重要性的流程相似性算法 [J]. 计算机集成制造系统，2015, 21 (2): 344-358. (Yin Ming, Wen Lijie, Wang Jianmin, *et al.* Process similarity algorithm based on importance of transition adjacent relations [J]. Computer integrated manufactu-ring system, 2015, 21 (2): 344-358.)

[6]王子璇，闻立杰，汪抒浩，等. 基于变迁标签图编辑距离的过程模型相似性度量 [J]. 计算机集成制造系统, 2016, 22 (2): 343-352. (Wang Zixuan, Wen Lijie, Wang Shuhao, *et al.* Similarity measurement for process models based on transition-labeled graph edit distance [J]. Computer integrated manufacturing system, 2016, 22 (2): 343-352.)

[7]Zha Haiping, Wang Jianmin, Wen Lijie, *et al.* A workflow net similarity measure based on transition adjacency relations [J]. Computers in Industry, 2010, 61 (5): 463-471.

[8]Weidlich M, Elliger F, Weske M. Generalised computation of behavioural profiles based on Petri-net unfoldings [M]// Web Services and Formal Methods. Berlin: Springer, 2010: 101-115.

[9]Dijkman R, Dumas M. Graph matching algorithms for business process model similarity search [C]// Proc of International Conference on Business Process Management. [S.l.]:Springer-Verlag, 2009: 48-63.

[10]Dijkman R, Dumas M, Dongen B V, *et al.* Similarity of business process models: metrics and evaluation [J]. Information Systems, 2011, 36 (2): 498-516.

[11]Jin Tao, Wang Jianmin, Wen Lijie. Efficiently querying business process models with beehiveZ [C]// Proc of Demo Track of the 9th Conference on Business Process Management.2011.

[12]武年华，金涛，查海平，等. BeehiveZ: 一个开放的业务过程模型管理框架 [J]. 计算机研究与发展, 2010, 47 (z1): 450-454. (Wu Nianhua, Jin Tao, Cha Haiping, *et al.* BeehiveZ: an open business process model management framework. [J]. Computer Research and Development, 2010, 47 (z1): 450-454.)

[13]Dong Zihe, Wen Llijie, Huang Haowei, *et al.* CFS: a behavioral similarity algorithm for process models based on complete firing sequences [J]. Journal of Software, 2015: 202-219.